OPIC
OFFICE DE LA PROPRIÉTÉ INTELLECTUELLE DU CANADA

CIPO
CANADIAN INTELLECTUAL PROPERTY OFFICE

(12) (19) (CA) **Demande-Application**

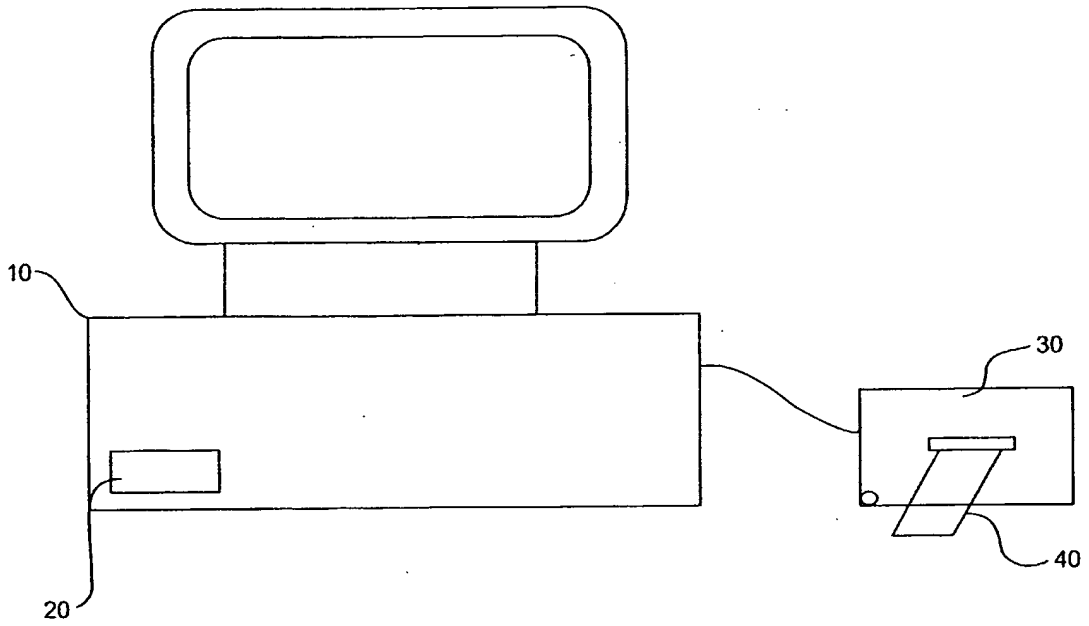
(21) (A1) **2,267,041**
(22) 1999/03/26
(43) 2000/09/26

(72) BORZA, Stephen J., CA
(71) DEW ENGINEERING AND DEVELOPMENT LIMITED, CA

(51) Int.Cl.[6] G06F 12/14, G06K 19/077, G06F 9/00

(54) **METHODE ET SYSTEME DE PROTECTION CONTRE LE PIRATAGE D'UNE APPLICATION LOGICIELLE**

(54) **METHOD AND SYSTEM FOR PROTECTING A SOFTWARE APPLICATION FROM PIRACY**



(57) A method and a system for disabling execution of a software application stored within a computer absent data indicative of an authorised use of the software application are disclosed. At start up or during execution of a software application a user is prompted for user authorisation information. Using a processor within a smart card the received user authorisation information is compared with user authorisation information stored in memory of the smart card to produce a comparison result. If the comparison result is indicative of an authorised user of the software application, then data indicative of the authorised use of the software application is provided from the smart card to the computer. Upon receipt of the data indicative of the authorised use of the software application execution of the software application is continued. When the data is not data indicative of the authorised use of the software application further execution of the software application is disabled. For example, biometric information in the form of a fingerprint may be used for user authorisation.

Abstract of the Disclosure

A method and a system for disabling execution of a software application stored within a computer absent data indicative of an authorised use of the software application are disclosed. At start up or during execution of a software application a user is prompted for user authorisation information. Using a processor within a smart card the received user authorisation information is compared with user authorisation information stored in memory of the smart card to produce a comparison result. If the comparison result is indicative of an authorised user of the software application, then data indicative of the authorised use of the software application is provided from the smart card to the computer. Upon receipt of the data indicative of the authorised use of the software application execution of the software application is continued. When the data is not data indicative of the authorised use of the software application further execution of the software application is disabled. For example, biometric information in the form of a fingerprint may be used for user authorisation.

Method and System for Protecting a Software Application from Piracy

Field of the Invention

This invention relates generally to a method for disabling execution of a software application stored within a computer absent data indicative of an authorised use of the software application and more particularly relates to a method for disabling execution of a software application using a smart card.

Background of the Invention

Software piracy is causing huge losses in profits for companies developing and selling computer software. The sale and distribution of software designed for general use is based on the assumption that a software provider only wishes to sell a right to use the software to one person or to a select group of people. The seller specifically does not wish the purchaser to distribute the software to other users or to resell the software. Various methods are employed to prevent the purchaser from distributing the software. The most common method of software distribution is via a storage medium. The user is often unrestricted from either copying the software from this storage medium to another one or installing the software from a same storage medium on different computers. This results in undesirable software piracy.

In the past, piracy protection schemes were implemented to prevent a purchaser from making a back up copy of a software application. When a storage medium, such as a floppy disk, that is copy protected becomes damaged, the software provider must replace the storage medium. Some software providers have included a backup copy of the software application along with the original copy to lessen inconvenience when the original copy is damaged, but sometimes even these "backup" copies fail.

Some software providers have avoided software copy protection schemes. Instead these software providers rely on the honesty of the purchaser, the fact that the documentation is difficult to duplicate, and/or a license agreement that the purchaser is expected to honour. The license agreement makes it illicit to distribute the software.

Many software providers view convenience as essential for software users and therefore are wary of software piracy protection methods that inconvenience users in any way.

Another method of software protection involves writing to an installation disk to indicate that installation has occurred. This limits a disk to one use. This has many of the aforementioned drawbacks and also is unworkable with CD ROM technology. Using CD ROM technology, a software provider writes a program and other information to a CD ROM which can be written to only once using special hardware for that purpose. Therefore, adding information to a CD ROM during installation is not possible.

With the increasing use of digital communications such as the Internet, computer software is now commonly distributed using these means. In this case, the aforementioned methods of preventing undesired proliferation of pirated software can not be applied; users require some form of backup and this backup is easily distributed to other users. Also, interception of software by unauthorised third parties is a significant risk to software providers .

Computer security is fast becoming an important issue. With the proliferation of computers and computer networks into all aspects of business and daily life - financial, medical, education, government, and communications - the concern over secure file access is growing. Using passwords is a common method of providing security. Password protection is employed for computer network security, automatic teller machines, telephone banking, calling cards, and telephone answering services. These systems generally require knowledge of an entry code that has been selected by a user or has been configured in advance. Examples of commonly used security codes for preventing software piracy include information from a user's manual and a serial number. Unfortunately for use in copy protection, security codes are unworkable since the software is easily transferred with the security code.

A security access system that provides substantially secure access and does not require a password or access code is a biometric identification system. A biometric identification system accepts unique biometric information from a user and identifies the

user by matching the information against information belonging to registered users of the system. One such biometric identification system is a fingerprint recognition system.

The use of a biometric imaging device with a personal computer is becoming widespread. In a fingerprint input transducer or sensor, the finger under investigation is usually pressed against a flat surface, such as a side of a glass plate; the ridge and valley pattern of the finger tip is sensed by a sensing means such as an interrogating light beam. Various optical devices are known which employ prisms upon which a finger whose print is to be identified is placed. The prism has a first surface, a platen, upon which a finger is placed, a second surface disposed at an acute angle to the first surface through which the fingerprint is viewed and a third illumination surface through which light is directed into the prism. In some cases, the illumination surface is at an acute angle to the first surface, as seen for example, in US Patents 5,187,482 and 5,187,748. In other cases, the illumination surface is parallel to the first surface, as seen for example, in US Patents 5,109,427 and 5,233,404. Fingerprint identification devices of this nature are generally used to control the building-access or information-access of individuals to buildings, rooms, and devices such as computer terminals.

United States patent number 4,353,056 in the name of Tsikos issued October 5, 1982, discloses an alternative kind of fingerprint sensor that uses a capacitive sensing approach. The described sensor has a two dimensional, row and column, array of capacitors, each comprising a pair of spaced electrodes, carried in a sensing member and covered by an insulating film. The sensors rely upon deformation to the sensing member caused by a finger being placed thereon so as to vary locally the spacing between capacitor electrodes, according to the ridge/trough pattern of the fingerprint, and hence, the capacitance of the capacitors. In one arrangement, the capacitors of each column are connected in series with the columns of capacitors connected in parallel and a voltage is applied across the columns. In another arrangement, a voltage is applied to each individual capacitor in the array. Sensing in the respective two arrangements is accomplished by detecting the change of voltage distribution in the series connected capacitors or by measuring the voltage values of

the individual capacitances resulting from local deformation. To achieve this, an individual connection is required from the detection circuit to each capacitor.

Before the advent of computers and imaging devices, research was conducted into fingerprint characterisation and identification. Today, much of the research focus in biometrics has been directed toward improving the input transducer and the quality of the biometric input data. Fingerprint characterisation is well known and can involve many aspects of fingerprint analysis. The analysis of fingerprints is discussed in the following references, which are hereby incorporated by reference:

Xiao Qinghan and Bian Zhaoqi,: An approach to Fingerprint Identification By Using the Attributes of Feature Lines of Fingerprint," IEEE Pattern Recognition, pp 663, 1986;

C.B. Shelman, " Fingerprint Classification - Theory and Application," Proc. 76 Carnahan Conference on Electronic Crime Countermeasures, 1976;

Feri Pernus, Stanko Kovacic, and Ludvik Gyergyek, "Minutaie Based Fingerprint Registration," IEEE Pattern Recognition, pp 1380, 1980;

J.A. Ratkovic, F.W. Blackwell, and H.H. Bailey, "Concepts for a Next Generation Automated Fingerprint System," Proc. 78 Carnahan Conference on Electronic Crime Countermeasures, 1978;

K. Millard, " An approach to the Automatic Retrieval of Latent Fingerprints," Proc. 75 Carnahan Conference on Electronic Crime Countermeasures, 1975;

Moayer and K.S. Fu, "A Syntactic Approach to Fingerprint Pattern Recognition," Memo Np. 73-18, Purdue University, School of Electrical Engineering, 1973;

Wegstein, *An Automated Fingerprint Identification System*, NBS special publication, U.S. Department of Commerce/National Bureau of Standards, ISSN 0083-1883; no. 500-89, 1982;

Moenssens, Andre A., Fingerprint Techniques, Chilton Book Co., 1971; and,

Wegstein and J.F. Rafferty, *The LX39 Latent Fingerprint Matcher*, NBS special publication, U.S. Department of Commerce/National Bureau of Standards; no. 500-36, 1978.

Object of the Invention

It is an object of this invention to disable execution of a software application stored within a computer absent data indicative of an authorised use of the software application using a smart card.

Statement of the Invention

In accordance with the invention there is provided a method for protecting a software application from piracy comprising the steps of:
providing data indicative of an authorised use of the software application;
executing a first portion of the software application;
receiving user authorisation information;
using a processor within a peripheral device, comparing the received user authorisation information with user authorisation information stored in memory of the peripheral device to produce a comparison result and, if the comparison result is indicative of the authorised user of the software application, providing data from the peripheral device to the computer, the data indicative of the authorised use of the software application on the computer; and,
executing a second portion of the software application only upon receipt of the data indicative of the authorised use of the software application.

In accordance with the invention there is also provided a method for protecting a software application from piracy comprising the steps of:
providing data indicative of an authorised use of the software application;
executing a first portion of the software application using a first processor;
receiving user authorisation information from an input device disposed within a first housing;
transmitting the user authorisation information to a second processor within a second housing, the second housing comprising memory, wherein the user authorisation information is unavailable to the first processor;

using the second processor, comparing the received user authorisation information with user authorisation information stored in memory to produce a comparison result and, if the comparison result is indicative of the authorised user of the software application, providing data from the second processor to the first processor, the data indicative of the authorised use of the software application on the first processor; and,
executing a second portion of the software application only upon receipt of the data indicative of the authorised use of the software application.

In accordance with another aspect of the invention there is provided a system for protecting a software application from piracy comprising:
a software application for execution on a computer, the software application including executable commands for preventing execution of a portion of the software application absent data indicative of an authorised use of the software application; and
a peripheral device including:
a housing;
a port for interfacing the peripheral device to the computer;
memory within the housing, the memory for storing user authorisation information indicative of an authorised user of the software application; and,
a processor within the housing for receiving authorisation information provided by a user, for comparing received authorisation information with the user authorisation information stored in the memory to produce a comparison result and, if the comparison result is indicative of an authorised user of the software application, providing data from the device to the computer, the data indicative of an authorised use of the software application on the computer,
wherein the software application execution is at least partially prevented until the data indicative of the authorised use of the software application is received.

Preferably the peripheral device comprises a smart card reader and a smart card. More preferably, the peripheral device also comprises a biometric sensor in the form of a fingerprint imager.

Brief Description of the Drawings

An exemplary embodiment of the invention will now be described in conjunction with the attached drawings, in which:

Fig. 1a is a simplified diagram of a system according to the invention for protecting a software application from piracy using a smart card;

Fig. 1b is a simplified flow diagram of a method according to the invention for protecting a software application from piracy using the system described in Fig. 1a;

Fig. 2a is a simplified diagram of a system according to the invention for protecting a software application from piracy storing a serial number of a processor memory of a smart card;

Fig. 2b is a simplified flow diagram of an installation of a software application using the system described in Fig. 2a;

Fig. 2c is a simplified flow diagram of an execution of a software application using the system described in Fig. 2a ;

Fig. 3a is a simplified diagram of a preferred embodiment of a system according to the invention for protecting a software application from piracy comprising a smart card reader with a contact imager disposed therein;

Fig. 3b is a simplified flow diagram of a method according to the invention for protecting a software application from piracy using the system described in Fig. 3a;

Fig. 4a is a simplified diagram of a system according to the invention for protecting a software application from piracy comprising a smart card reader and a contact imager connected to a computer;

Fig. 4b is a simplified diagram of a system according to the invention for protecting a software application from piracy comprising a smart card with a contact imager;

Fig. 5 is a simplified flow diagram of a method according to the invention for protecting a software application from piracy using one smart card for different software applications;

Fig. 6 is a simplified flow diagram of a method according to the invention for protecting a software application from piracy using one smart card for different users;

Fig. 7a is a simplified flow diagram of a method according to the invention for protecting a software application from piracy using different versions of the software application for different users;

Fig. 7b is a simplified diagram of a computer network for implementing the method described in Fig. 7a; and,

Fig. 8 is a simplified flow diagram of a method according to the invention for protecting a software application from piracy storing additional data on the smart card.

Detailed Description

With the advent of personal computers, software piracy – copying software in which copyright exists – proliferated. With the implementation of graphical user interfaces, copying software has become a matter of dragging files from an original diskette to a blank diskette. In fact, software piracy is so prolific that it is estimated that more than one half of software in use today is pirated. Obviously, it is in the best interests of software providers to reduce software piracy.

In the specification and claims that follow all items peripheral to computer functionality such as disk drives including hard disk drives, smart card readers, scanners, keyboards, printers, imaging devices, etc. are referred to as peripheral devices.

The invention provides a unique method for distribution of piracy-protected software. The method relies on a peripheral device such as a smart card to unlock software in order to permit execution thereof. The smart card is a credit card sized electronic device comprising memory, a processor and an interface. The processor provides for predetermined and/or flexible execution of software within the smart card. The memory comprises RAM for use during software execution and ROM for long term storage of information. Preferably, some electrically erasable ROM is provided to allow for reprogramming of the smart card. A smart card is a suitable storage medium for

storing data related to an authorised use of a software application. The data are stored in ROM within the smart card and the ROM are only accessible to the software application. Such a smart card is immutable and therefore, prevents software piracy. As such, only an individual with the smart card for a particular software package may execute the software. In an embodiment, each software application is customised to operate with only one unique smart card thereby restricting use of the application to systems in communication with the one unique smart card.

In a further embodiment, the smart card is provided with security such as a password or biometric information verification to prevent theft of the software or unauthorised use thereof. Though the method according to the invention is described with reference to fingerprint registration, it is applicable to other biometric information and methods of user authentication using that biometric information. An individual's biometric information in the form of a fingerprint, retinal scan, palm print, voice print, etc. from a biometric information source in the form of a fingertip, eye, hand, voice, etc. is captured using a biometric information input device in the form of a contact imager, eye scanner, or a microphone. Contact imager, as described above, are known in the art of electronic security. Using a contact imager, a fingerprint is digitised and, using a processor, the digitised fingerprint is characterised. Characterisation of fingerprints is known in the art of fingerprint analysis and, in general involves extracting features in a substantially global reference frame for comparison against template features.

The characterised digitised biometric information is registered with stored templates to locate a template that, within a predetermined level of security in the form of a predetermined false acceptance rate, matches the characterised information. When a match is detected, the individual is identified and a corresponding action is performed. Alternatively, when identification is not necessary the individual is authenticated. Some known actions performed based on biometric authentication include system access, unlocking a door, logging access, providing personal information, billing an individual, etc.

Referring to Figs. 1a and 1b, a simplified diagram of a method to protect a software application from piracy according to the invention is shown. A software provider during packaging customises a software application **20** to a particular smart card **40**. The smart card **40** and the software application **20** are then packaged together for sale. The smart card **40** and the software application **20** within the package function to provide a purchaser with operational software. Absent the smart card **40**, the software application **20** will not function.

Such a software package is not easily pirated. Either the software instructions requiring the presence of the smart card are removed, or the smart card is duplicated. Neither of these is easily accomplished. Therefore, simple drag and drop piracy is prevented greatly reducing the amount of piracy. Further, pirated copies are more easily detected for enforcement of copyright in software.

The software application **20** is installed on a computer **10** to which a smart card reader **30** is connected. In order to execute the software application **20** the smart card **40** is inserted into the smart card reader **30** in order to validate an authorised use of the software. The validation occurs during start up of the software application **20** or every so often during start up or normal use. The validation merely verifies a presence of the smart card **40** associated with the software application **20**. When the correct smart card **40** is not present, the user is prompted to insert the correct smart card **40** and software execution is paused until the correct smart card **40** is detected.

Figs. 2a, 2b and 2c show a simplified diagram of another method according to the invention. During installation of a software application **20** a serial number of a processor **15** for executing the software application **20** is written on a write once ROM of a smart card **40** which is provided together with the software. During an initialisation of the software application **20** or at intervals during execution of the software application **20** the authorised use of the software application **20** is validated. The validation verifies a presence of the smart card **40** and the serial number of the processor **15**. Providing secure access to the serial number of a processor is known to those of skill in the art, for

example as disclosed in U.S. Patent 5,790,663 issued Aug. 4, 1998 to Lee et al.. Writing a serial number of a processor on a write once ROM of a smart card limits the execution of a software application to one processor. In case a user wants to resell the software application or wants to change the processor for executing the software application he would contact a software provider for a new smart card. Therefore, this method provides a software provider with the ability to control distribution of a software application by limiting the execution of the software application to one processor, while allowing a user to resell software or change the processor on which the software is executed. The difficulties involved in removing instructions requiring a presence of the smart card or the inherent difficulties of copying the smart card substantially prevent piracy of the software application. Writing a serial number of a processor into a software application upon installation is disclosed in U.S. Patent 5,790,663. Unfortunately, this still allows copying and distribution of a software application prior to installation because the piracy protection - the serial number - is yet to be stored within the application.

Referring to Figs. 3a and 3b, a preferred embodiment of a method for protecting a software application from piracy according to the invention is shown. The software application **20** is installed on a computer **10**. A smart card reader **50** is connected through a first port **52** via a cable **70** to the computer **10**. A smart card **40** comprising a processor **42** and memory **44** is connected through a second port **54** to the smart card reader **50**. Furthermore, a contact imager **60** is disposed within the housing of the smart card reader **50**. For example, a smart card reader with a built in contact imager for use with a personal computer is available from American Biometric Corporation® under the product name the Biomouse Plus®. During execution, the software application **20** requires that data indicative of an authorised use be provided from the smart card **40**. Upon receipt of a request for the data indicative of an authorised use a user is prompted for fingerprint information. The user provides the fingerprint information using the contact imager **60**. The fingerprint information is provided to the smart card **40**. Preferably, the information is not accessible to the computer **10**. The fingerprint information is compared with fingerprint information stored in the memory **44** of the smart card **40** using the processor

42. If the comparison result is indicative of an authorised user of the software application, data indicative of the authorised use in the form of a key to unlock the software application is provided to the computer **10**. Using a processor **15** of the computer, the received data is validated. Validating the data ensures that only a smart card **40** for use with the software application is used. Optionally, validation includes decryption of the data using a known decryption key. The method according to the invention is highly advantageous compared to prior art methods from the point of view of a software provider as well as a user. It provides additional security by storing data related to user authorisation such as a password or, preferably, biometric information within a peripheral device for use in enabling and disabling software execution. Use of biometric information is preferred because it is a substantially secure form of user authorisation. Storage of data related to user authorisation enables a software provider to require user registration prior to installation of the software. Therefore, this method provides a software provider with a substantially secure method to control distribution of a software application and to protect it from piracy. Since a smart card is required to execute the software application, only those who purchase the smart card from the software provider have access to executable copies of the software. At the same time it provides benefits to a user by securing a software application from unauthorised use and additionally securing access to files, for example when the software application is needed to open a file.

Optionally, a processor is disposed within the smart card reader **50**. Fingerprint information is received from the contact imager **60** and compared with fingerprint information transmitted from the smart card **40** using the processor disposed within the smart card reader **50** to produce a comparison result. If the comparison result is indicative of an authorised user of the software application **20**, data indicating an authorised use of the software application are transmitted from the smart card **40** via the smart card reader **50** to the computer **10**.

Alternatively, security is further enhanced by encrypting the data indicating an authorised use of the software application **20** with asymmetric encryption using the processor **42** of the smart card **40** prior to the transfer to the computer **10**. Using the

processor **42** of the smart card **40** for encryption provides a secure communication link for transferring the data and allows alterations to the encryption key for each data transfer, thus preventing tampering with the data.

In another embodiment according to the invention the smart card reader **30** and the contact imager **60** are included in a computer pointing device such as a mouse.

Optionally, the smart card reader **50** and the contact imager **60** are disposed within a keyboard connected to the computer **10**.

Further optionally, the smart card reader **50** is included in a track pad, wherein the pointing device of the track pad is used as a capacitive contact imager to provide fingerprint information.

Fig. 4a shows another system for implementing the method according to the invention. A smart card reader **30** is connected to computer **10**. Furthermore, a contact imager **80** is connected to the computer **10**. Using this system for implementing a method according to the invention, biometric information received at the contact imager **80** is transferred from the contact imager **80** via the computer **10** to smart card **40**. Transferring the biometric information via the computer **10** enhances the risk of interference by a third party while the data is within the computer **10**.

The system shown in Fig. 4b comprises a smart card **45** comprising a capacitive contact imager **46**. The smart card **45** is connected to a computer **10** via a smart card reader **30**. The biometric information is captured and compared absent provision of the biometric data to the computer. A major drawback of this system is the high cost manufacturing a capacitive contact imager, thus making the smart card expensive.

Fig. 5 shows another embodiment of a method according to the invention, wherein one smart card is used for securing a plurality of software applications. During execution of a software application, data related to the software application and biometric information relating to a user of the software are provided to a smart card along with a request for data indicative of an authorised use of the software application. After receipt

of the request, the biometric information and the data, the biometric information is compared with biometric information stored within the smart card. If the comparison is indicative of an authorised user of the software application, data indicating an authorised use of the software application is selected based on the data provided to the smart card from a plurality of data indicating an authorised use for different software applications. This method enables a user to execute several software applications of a same software provider using only one smart card. It is highly advantageous for simultaneous execution of several software applications.Simultaneous execution of applications is supported by many common computer operating systems.

In Fig. 6 a method according to the invention is shown, wherein biometric information of a plurality of users is stored on one smart card. The method is similar to those set out above except that upon receipt of the biometric information, it is compared with the stored biometric information of the plurality of users. If the biometric information is indicative of an authorised user of the software application from the plurality of authorised users, data indicating an authorised use are provided to the computer. This method is highly advantageous when a plurality of users have access to a same computer, for example employees working at a same computer at different times during a day or system administrators in a corporation who wish to have access to all applications on all computers. A same smart card is left inserted in a smart card reader connected to the computer permitting execution of the software application to authorised employees.

Fig. 7a shows yet another embodiment of a method according to the invention. Different versions of a software application require different data indicating an authorised use of the software application. Each user is provided with a smart card having data indicative of an authorised use stored in memory. The data relates to a specific version of the software application. At start up and at intervals during execution of the software application, data indicating an authorised use is provided by a user of the software. The data is in the form of biometric information of the user. After receipt and comparison of the biometric information, data indicating an authorised use is provided to a computer in

execution of the software application. The data, once receivedis used to determine a version of the software application for execution. Alternatively, if the provided data do not indicate an authorised use of a current version of the software application further execution of the version is disabled.

Referring to Fig. 7b, a simplified diagram of a computer network **700** is shown. A plurality of workstations **704**, each comprising a personal computer, are connected by cables **706** to a server **702**. The workstations **704** are computer systems for use by users. The server **702** is for providing central file storage, network applications, and more powerful processors for processor intensive tasks. Some common network applications include email, Intranet, employee phone list, etc. Other network peripherals comprise printers, scanners, etc. (not shown). A system administrator maintains the computer network and authorises individual users thereof. Networks are well known in the art of computer systems and computer network architectures. They are installed in many businesses world-wide.

The method shown in Fig. 7a is highly advantageous for use with a computer network. Different versions of a software application **720** are stored on the server **702**. A user is permitted to use a predetermined version of the software. During start up of the software application, data indicative of an authorised use of a particular version are provided from the workstation **704** to the server **702**. According to the provided data the indicated version is selected and provided to the workstation **704** for execution. Alternatively, the indicated version of the application is executed on the server **702**. At intervals during execution of the software application, provision of data indicating an authorised use of the version is required and execution of the software application is interrupted when not provided. Therefore, according to this embodiment limited use of a software application according to employee work profile is permitted. In another embodiment of a method according to the invention shown in Fig. 8 additional data, comprising time varying data, are stored in the form of an indication of a maximum number of executions of a software application and a number indicating a current count of performed executions. At start up the two numbers are compared and if the number of

performed executions is equal to the maximum number of executions further execution of the software application is disabled. This method allows a software provider to rent a software application. A user renting a software application is provided with the software and a smart card. The smart card comprises data indicating the maximum number of executions. If the maximum number is reached and the user wants to continue renting the software application the software provider sends a new smart card or transfers a file to be stored on the smart card. Alternatively, the software is rented for a time period. In this case a time signal provided by a processor in execution of the software application is compared with time data stored in memory. Of course, it is preferred that the time data is provided by a timing circuit within the smart card in order to prevent clock tampering.

Alternatively, some of the additional data and some commands for execution on a processor are programmable by an authorised user. For example, a system administrator programs smart cards for use by employees of a company with a time limitation in order to prevent the employees from using the software application outside their work hours.

It is evident to those of skill in the art that prompting a user is realised in different ways depending on design criteria. One embodiment is to prompt the user for biometric information only during the installation of the software and to require a presence of a smart card during execution. This embodiment protects a software application from piracy and increases user convenience by prompting only once for biometric information but it allows execution of the software application to any user in possession of the smart card. Other methods include prompting a user for biometric information at the start-up of the software application and/or during the execution of the software application. Prompting during the execution of the software application is performed according to one of the following methods: one prompt at the beginning of software execution, several prompts at intervals during the use of the software application, and after breaks in use of the software application. Prompting the user several times during execution of a software application creates user inconvenience but may be desirable, for example in computer networks, to prevent unauthorised use of a software application on a workstation within the network . When user authorisation information in the form of biometric information is

to be provided frequently, transparent biometric data collection, i.e. the user is identified by collecting biometric information such as voice recognition, facial recognition, keystroke intervals etc., avoids the inconvenience of a plurality of pauses in software application execution and prompts during the pauses which increase user convenience.

As is evident to those of skill in the art, there are numerous methods to distribute a piracy protected software application according to the invention. For example a user sends biometric information to a software provider. The software provider then customises the software application by packaging the biometric information in a smart card and sends the software application together with the smart card to the user. The software application is executed only in presence of the smart card and the biometric information provided by the user. In another method a software provider sells the software application together with a smart card, the smart card comprising write once ROM. Upon installation, the user is prompted to store the biometric information in the smart card before completing same. Of course, when the smart card comprises rewritable ROM, the software application is transferable. It is evident to those of skill in the art, that because the smart card is needed to execute the software application, a number of executable copies of a software application is effectively limited to a number of interworking smart cards.

Alternatively, a software provider distributes a software application together with a smart card for a trial period and provides a user after registration with a registration file which is stored in the smart card.

Numerous other embodiments may be envisaged without departing from the spirit and scope of the invention.

Claims

What is claimed is:

1. A method for protecting a software application from piracy comprising the steps of:
providing data indicative of an authorised use of the software application;
executing a first portion of the software application;
receiving user authorisation information;
using a processor within a peripheral device, comparing the received user authorisation information with user authorisation information stored in memory of the peripheral device to produce a comparison result and, if the comparison result is indicative of the authorised user of the software application, providing data from the peripheral device to the computer, the data indicative of the authorised use of the software application on the computer; and,
executing a second portion of the software application only upon receipt of the data indicative of the authorised use of the software application.

2. A method for protecting a software application from piracy as defined in claim 1, comprising the step of:
when the data is not data indicative of the authorised use of the software application, disabling execution of a second portion of the software application.

3. A method for protecting a software application from piracy as defined in claim 1, comprising the steps of:
transmitting data related to the software application to the peripheral device; and,
comparing the received data with data stored in memory of the peripheral device using the processor of the peripheral device;
and wherein the step of providing data from the peripheral device comprises the step of:
selecting the data indicative of the authorised use of the software application on the computer according to the data related to the software application, the data selected from

a plurality of instances of data, each instance of data indicative of the authorised use of a different software application.

4. A method for protecting a software application from piracy as defined in claim 1, wherein the step of comparing the received user authorisation information with the user authorisation information stored in memory of the peripheral device to produce a comparison result comprises the step of:
comparing the provided user authorisation information with a plurality of instances of user authorisation information stored in memory.

5. A method for protecting a software application from piracy as defined in claim 4, wherein the step of providing data from the peripheral device comprises the step of:
selecting data indicative of the authorised use of the software application from a plurality of instances of data, the selection based on the user authorisation information provided.

6. A method for protecting a software application from piracy as defined in claim 5, comprising the step of:
selecting a version of the software application based on the data indicative of the authorised use of the software application received from the peripheral device.

7. A method for protecting a software application from piracy as defined in claim 1, comprising the steps of:
storing a first instance of data in memory of the peripheral device;
storing a second instance of data in memory of the peripheral device; and,
using the processor of the peripheral device, comparing the second instance of data with the first instance of data to produce a comparison result, and if the comparison result is indicative of an authorised use of the software application providing data indicative of the authorised use of the software application and calculate a new second instance of data to replace the second instance of data.

8. A method for protecting a software application from piracy as defined in claim 1, comprising the steps of:

receiving a time signal from the computer;

using the processor of the peripheral device, comparing the received time signal with time data stored in memory of the peripheral device to produce a comparison result, and if the comparison result is not indicative of an authorised use of the software application prevent at least partially execution of the software application.

9. A method for protecting a software application from piracy as defined in claim 1, wherein the peripheral device comprises a smart card.

10. A method for protecting a software application from piracy as defined in claim 1, wherein the user authorisation information comprises biometric information.

11. A method for protecting a software application from piracy as defined in claim 10, wherein the biometric information comprises fingerprint information.

12. A method for protecting a software application from piracy as defined in claim 1, wherein the data indicative of the authorised use of the software application are provided in an encrypted fashion.

13. A method for protecting a software application from piracy as defined in claim 1, wherein the processor of the peripheral device encrypts the data indicative of the authorised use of the software application.

14. A method for protecting a software application from piracy as defined in claim 13, wherein the data indicative of the authorised use of the software application are encrypted using asymmetric encryption.

15. A method for protecting a software application from piracy as defined in claim 1, wherein the user authorisation information is received during execution of the software application in intervals.

16. A method for protecting a software application from piracy comprising the steps of:
providing data indicative of an authorised use of the software application;
executing a first portion of the software application using a first processor;
receiving user authorisation information from an input device disposed within a first housing;
transmitting the user authorisation information to a second processor within a second housing, the second housing comprising memory, wherein the user authorisation information is unavailable to the first processor;
using the second processor, comparing the received user authorisation information with user authorisation information stored in memory to produce a comparison result and, if the comparison result is indicative of the authorised user of the software application, providing data from the second processor to the first processor, the data indicative of the authorised use of the software application on the first processor; and,
executing a second portion of the software application only upon receipt of the data indicative of the authorised use of the software application.

17. A method for protecting a software application from piracy as defined in claim 16, wherein the first housing comprises a biometric sensor and a smart card reader and wherein the second housing comprises a smart card.

18. A system for protecting a software application from piracy comprising:
a software application for execution on a computer, the software application including executable commands for preventing execution of a portion of the software application absent data indicative of an authorised use of the software application; and
a peripheral device including:
a housing;

a port for interfacing the peripheral device to the computer;
memory within the housing, the memory for storing user authorisation information indicative of an authorised user of the software application; and,
a processor within the housing for receiving authorisation information provided by a user, for comparing received authorisation information with the user authorisation information stored in the memory to produce a comparison result and, if the comparison result is indicative of an authorised user of the software application, providing data from the device to the computer, the data indicative of an authorised use of the software application on the computer,
wherein the software application execution is at least partially prevented until the data indicative of the authorised use of the software application is received.

19. A system for protecting a software application from piracy as defined in claim 18, wherein the housing comprises an input device for user authorisation information.

20. A system for protecting a software application from piracy as defined in claim 18 wherein the peripheral device comprises:
a first housing and a second housing, the first housing comprising:
the port for interfacing the peripheral device to the computer;
a second port within the first housing; and,
a transducer for receiving user authorisation information; and
the second housing comprising:
a port for interfacing with the second port of the first housing;
the memory; and,
the processor.

21. A system for protecting a software application from piracy as defined in claim 20, wherein the first housing comprises a smart card reader including a biometric sensor for sensing biometric information.

22. A system for protecting a software application from piracy as defined in claim 20, wherein the second housing comprises a smart card.

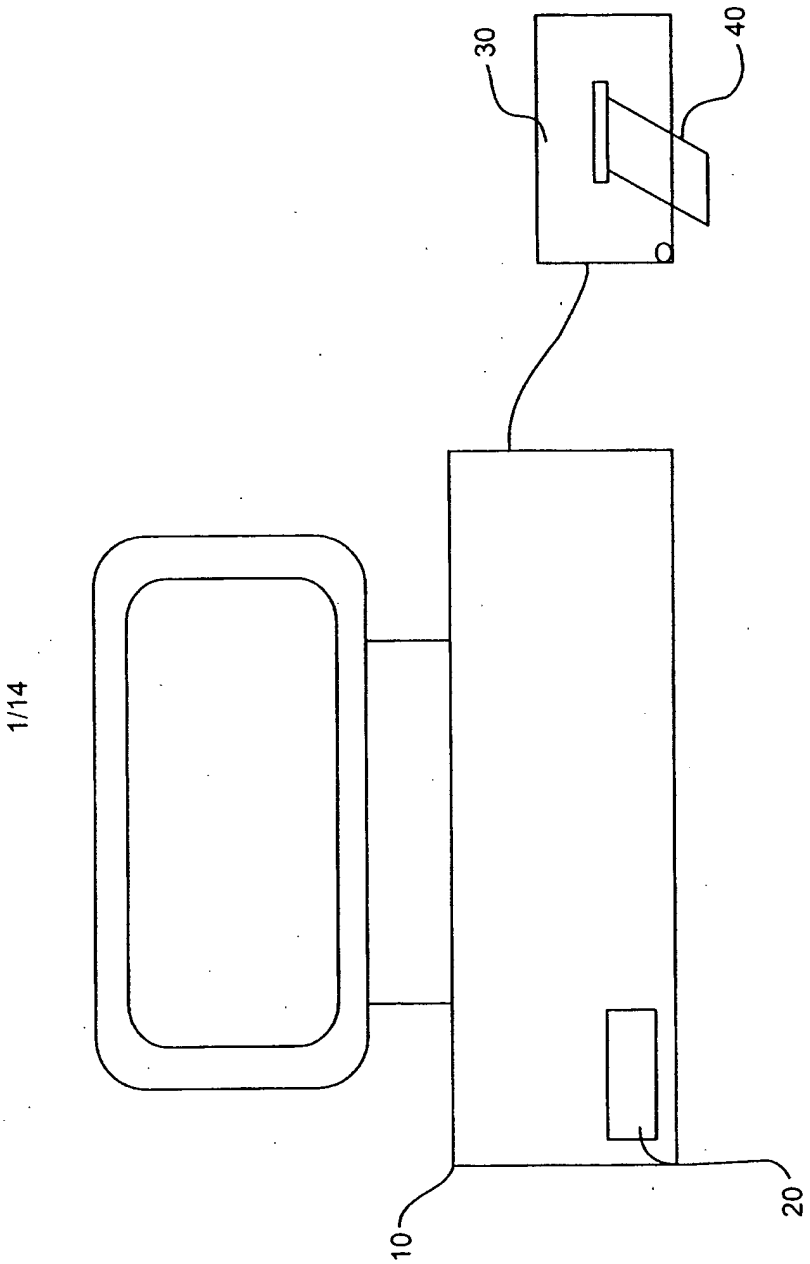


Figure 1a

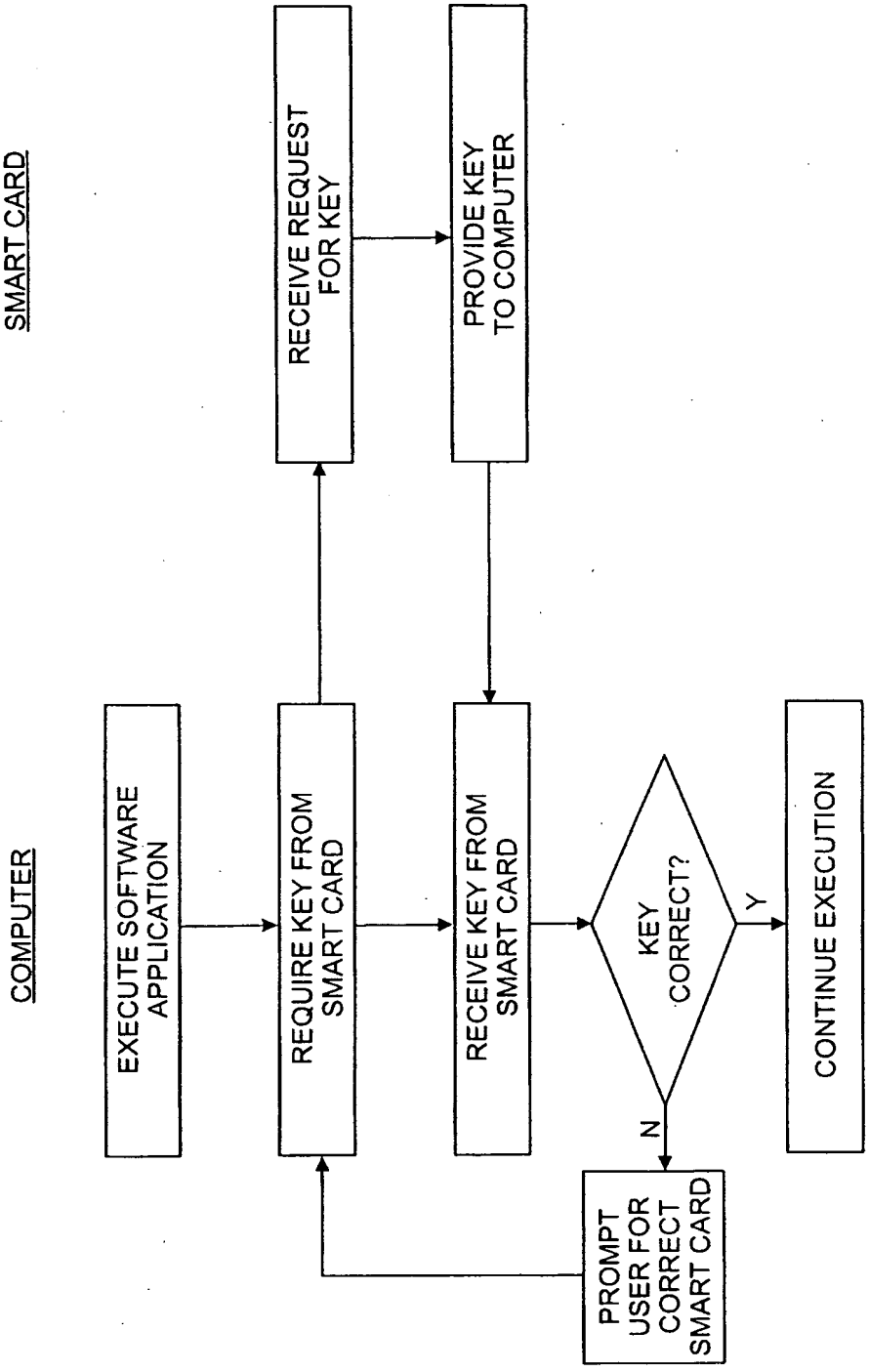

COMPUTER
SMART CARD
EXECUTE SOFTWARE APPLICATION
REQUIRE KEY FROM SMART CARD
RECEIVE REQUEST FOR KEY
PROVIDE KEY TO COMPUTER
RECEIVE KEY FROM SMART CARD
KEY CORRECT?
Y
N
CONTINUE EXECUTION
PROMPT USER FOR CORRECT SMART CARD

Figure 1b

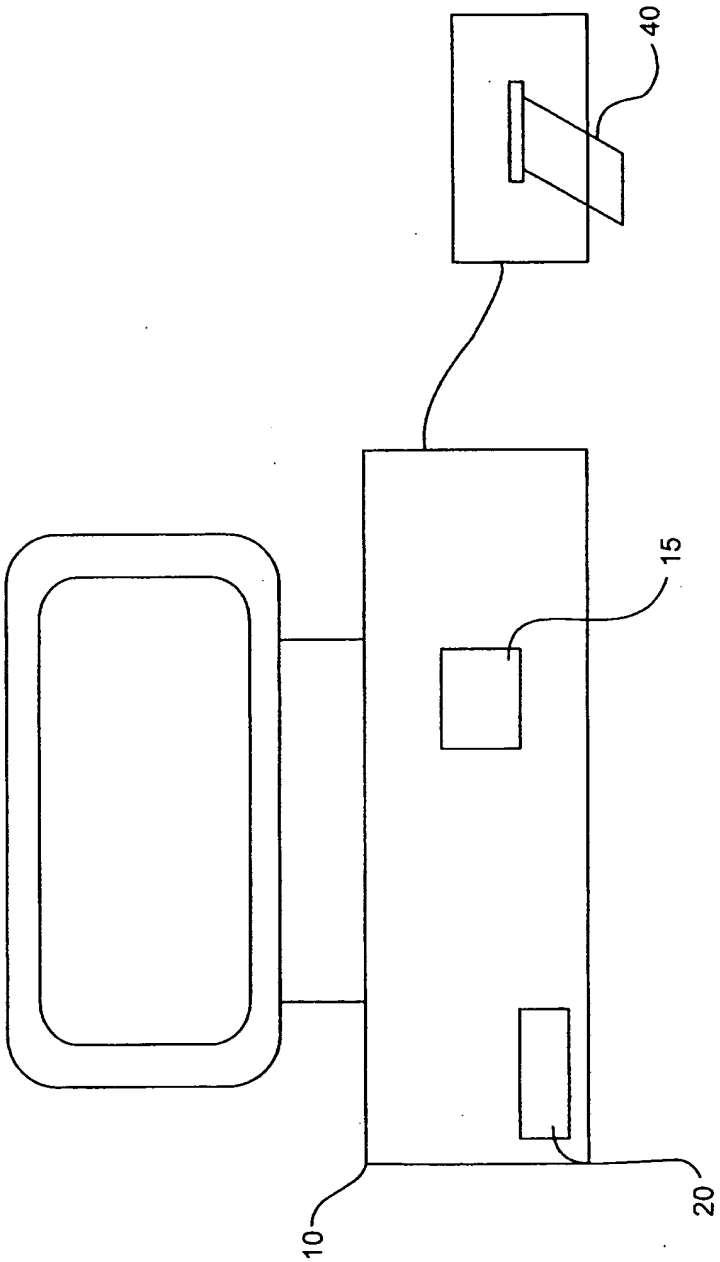


Figure 2a

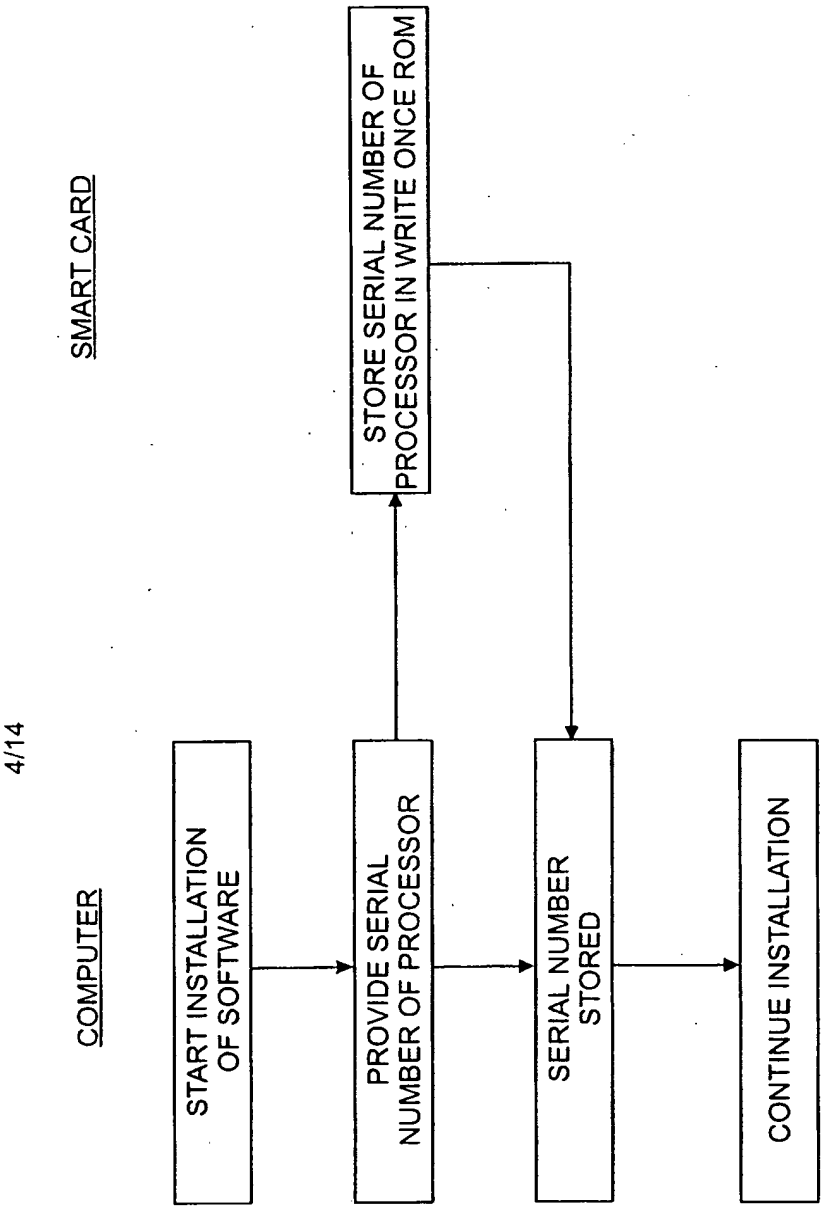

SMART CARD
STORE SERIAL NUMBER OF PROCESSOR IN WRITE ONCE ROM
COMPUTER
START INSTALLATION OF SOFTWARE
PROVIDE SERIAL NUMBER OF PROCESSOR
SERIAL NUMBER STORED
CONTINUE INSTALLATION

Figure 2b

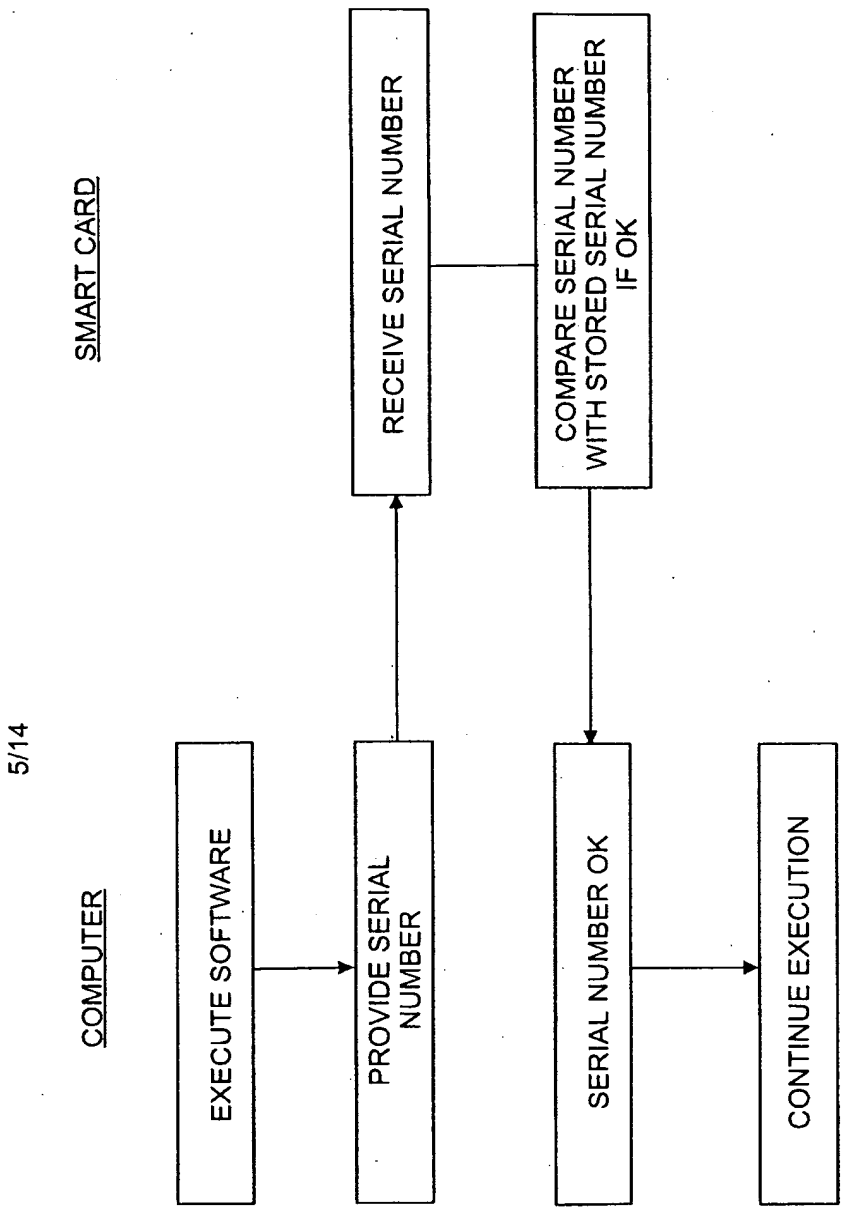

COMPUTER
SMART CARD
EXECUTE SOFTWARE
PROVIDE SERIAL NUMBER
RECEIVE SERIAL NUMBER
COMPARE SERIAL NUMBER WITH STORED SERIAL NUMBER IF OK
SERIAL NUMBER OK
CONTINUE EXECUTION

Figure 2c

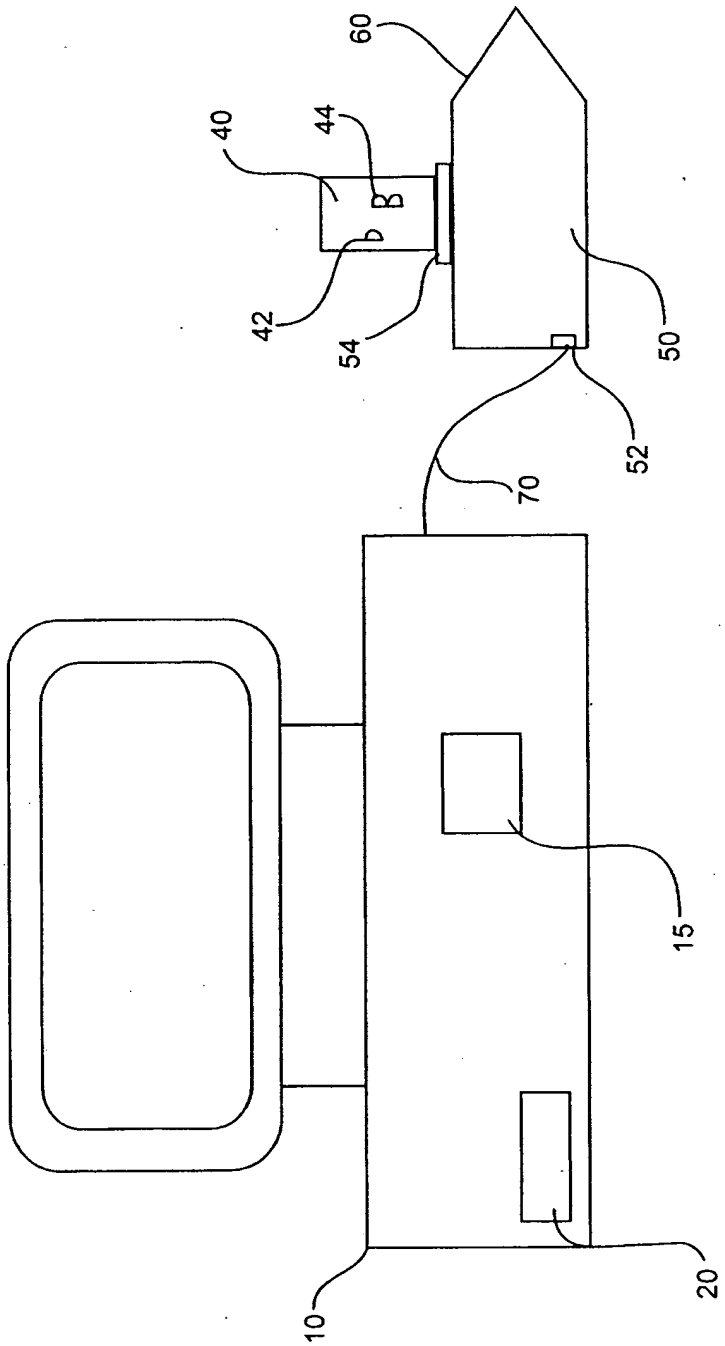


Figure 3a

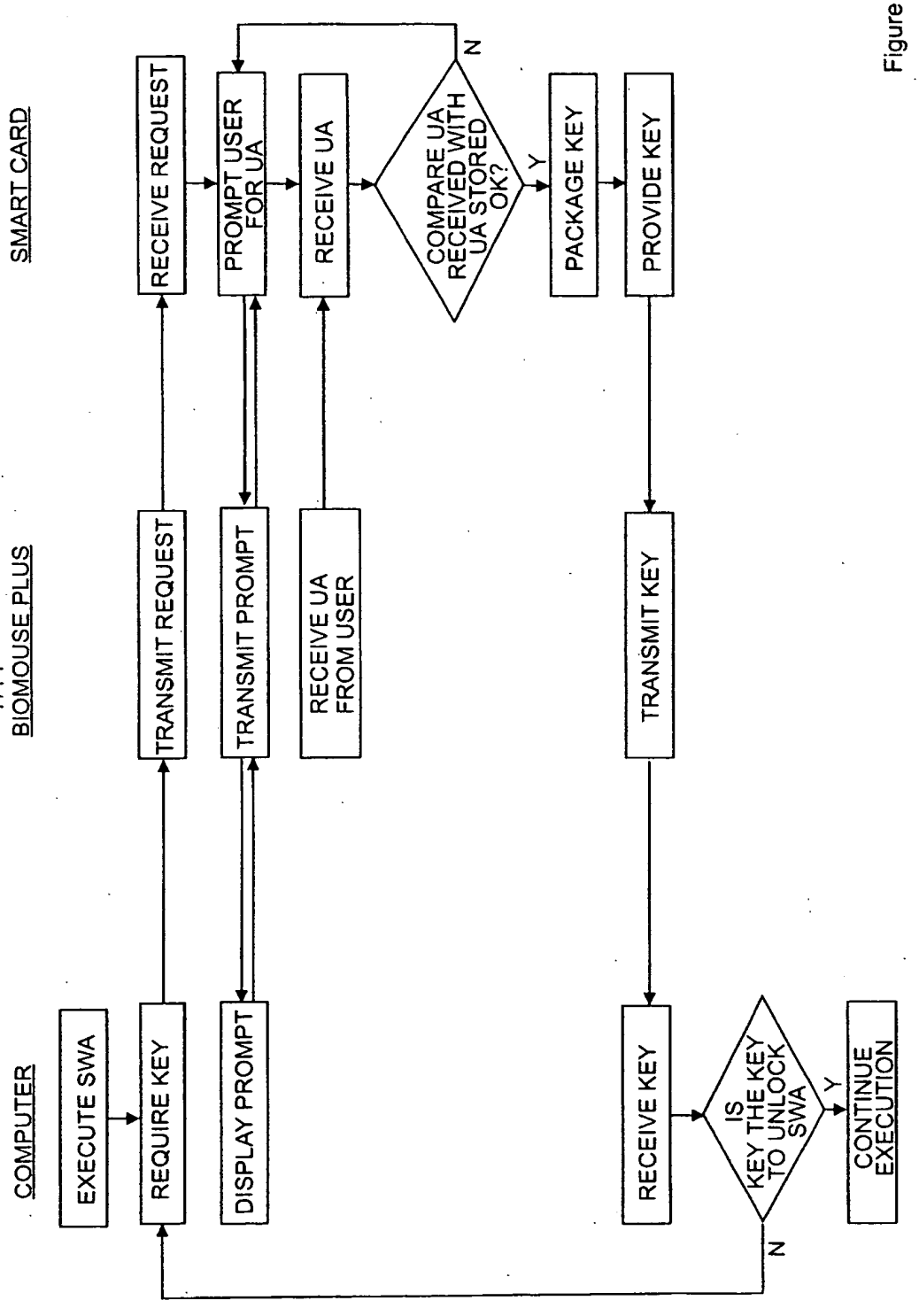

COMPUTER
7/14
BIOMOUSE PLUS
SMART CARD
EXECUTE SWA
REQUIRE KEY
TRANSMIT REQUEST
RECEIVE REQUEST
DISPLAY PROMPT
TRANSMIT PROMPT
PROMPT USER FOR UA
RECEIVE UA FROM USER
RECEIVE UA
COMPARE UA RECEIVED WITH UA STORED OK?
N
Y
PACKAGE KEY
PROVIDE KEY
TRANSMIT KEY
RECEIVE KEY
IS KEY THE KEY TO UNLOCK SWA
Y
N
CONTINUE EXECUTION

Figure 3b

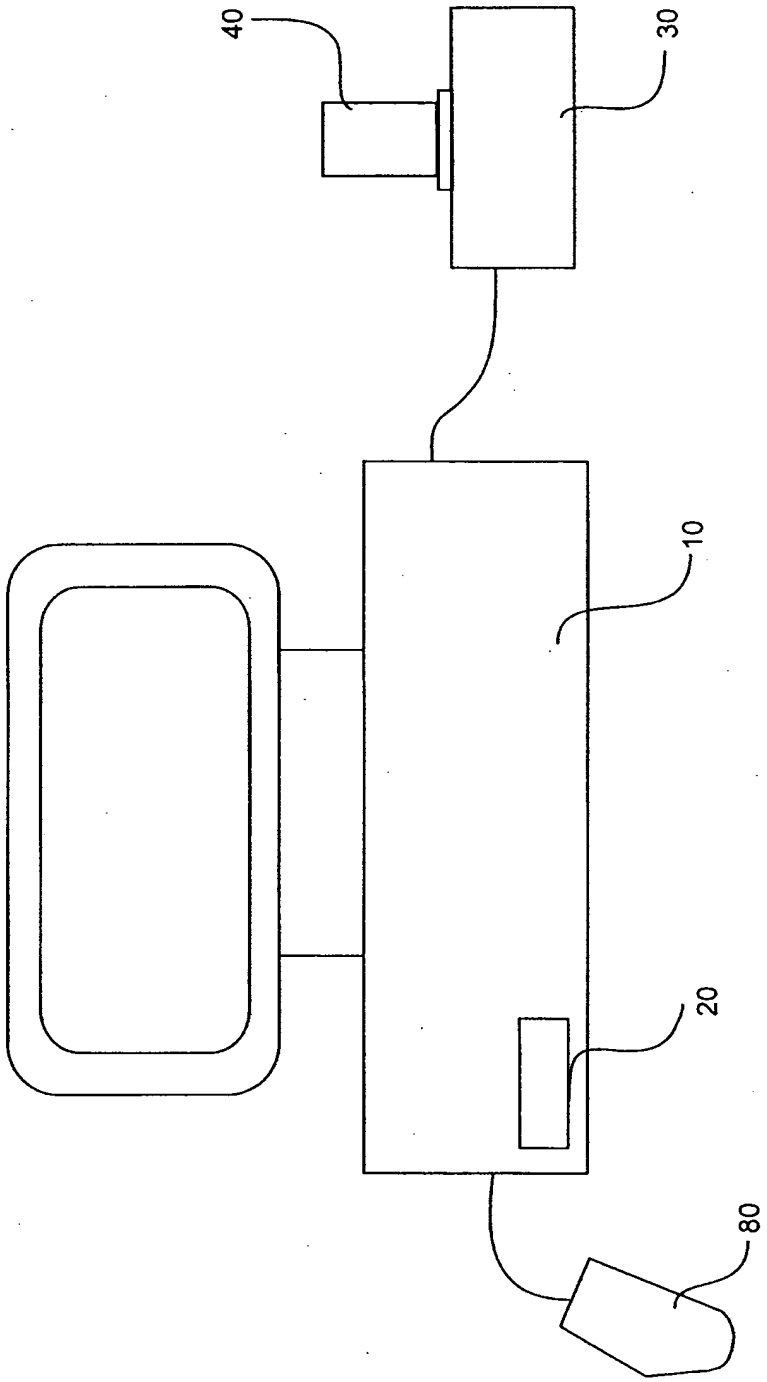


Figure 4a

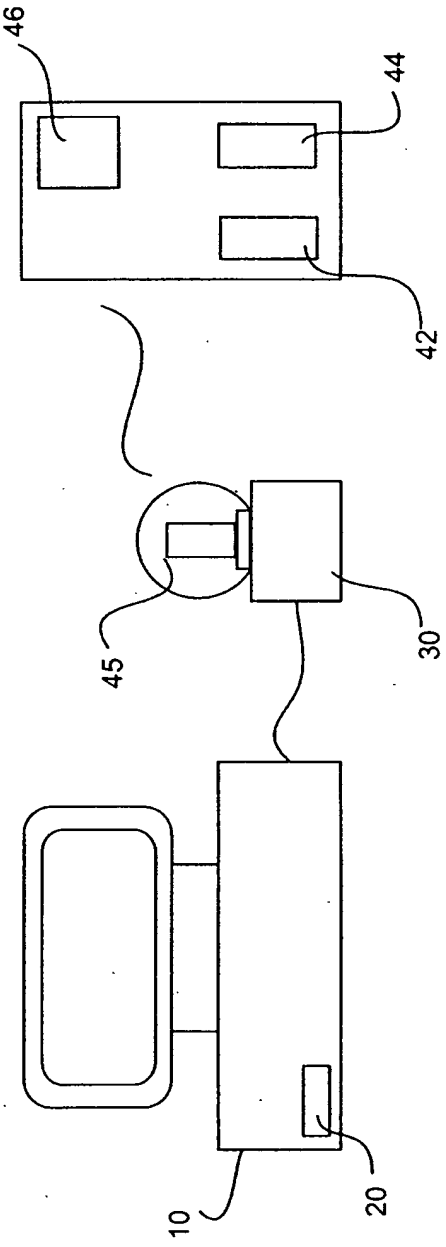


Figure 4b

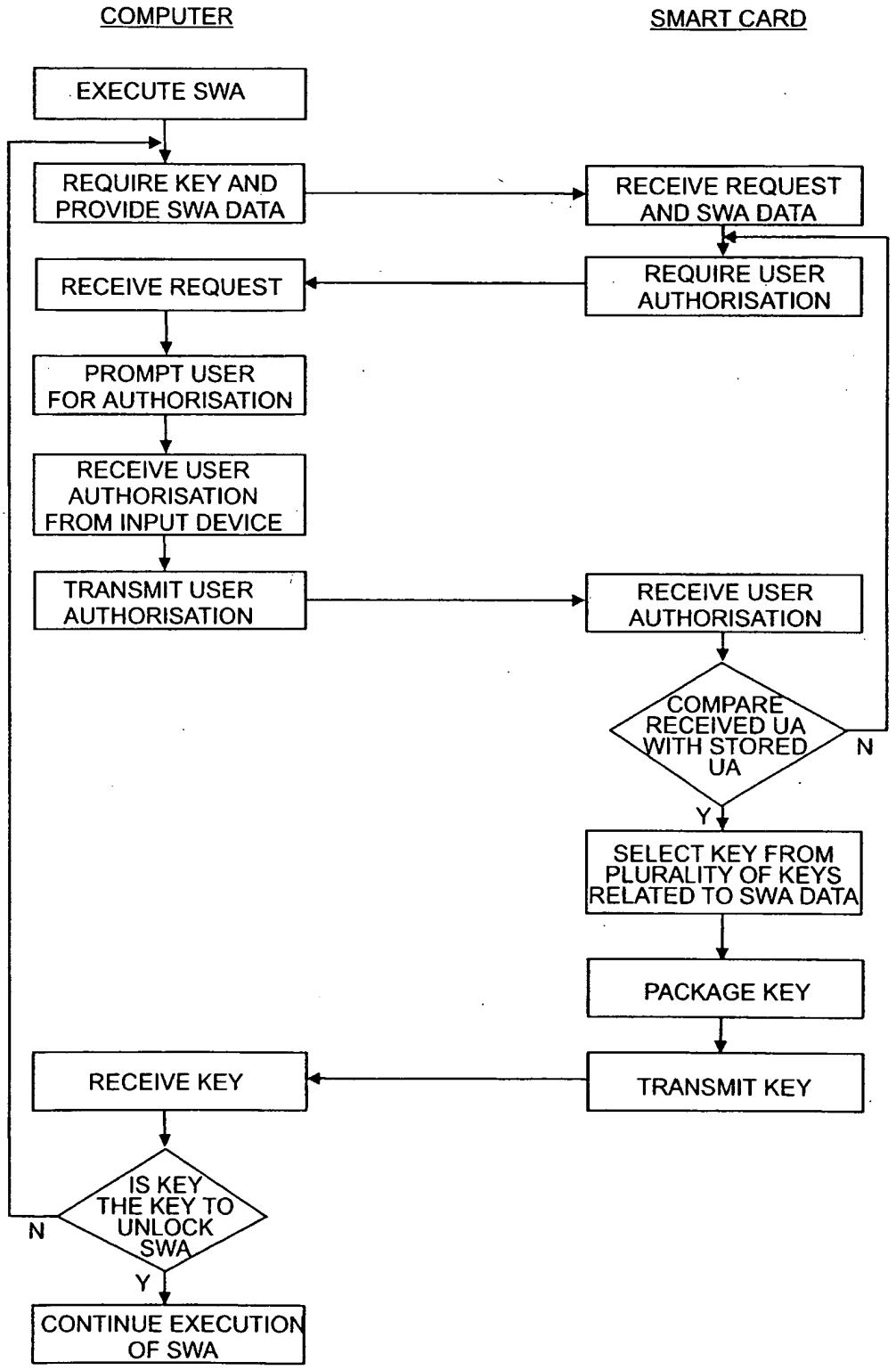

COMPUTER
SMART CARD
EXECUTE SWA
REQUIRE KEY AND PROVIDE SWA DATA
RECEIVE REQUEST AND SWA DATA
REQUIRE USER AUTHORISATION
RECEIVE REQUEST
PROMPT USER FOR AUTHORISATION
RECEIVE USER AUTHORISATION FROM INPUT DEVICE
TRANSMIT USER AUTHORISATION
RECEIVE USER AUTHORISATION
COMPARE RECEIVED UA WITH STORED UA
N
Y
SELECT KEY FROM PLURALITY OF KEYS RELATED TO SWA DATA
PACKAGE KEY
TRANSMIT KEY
RECEIVE KEY
IS KEY THE KEY TO UNLOCK SWA
N
Y
CONTINUE EXECUTION OF SWA

Figure 5

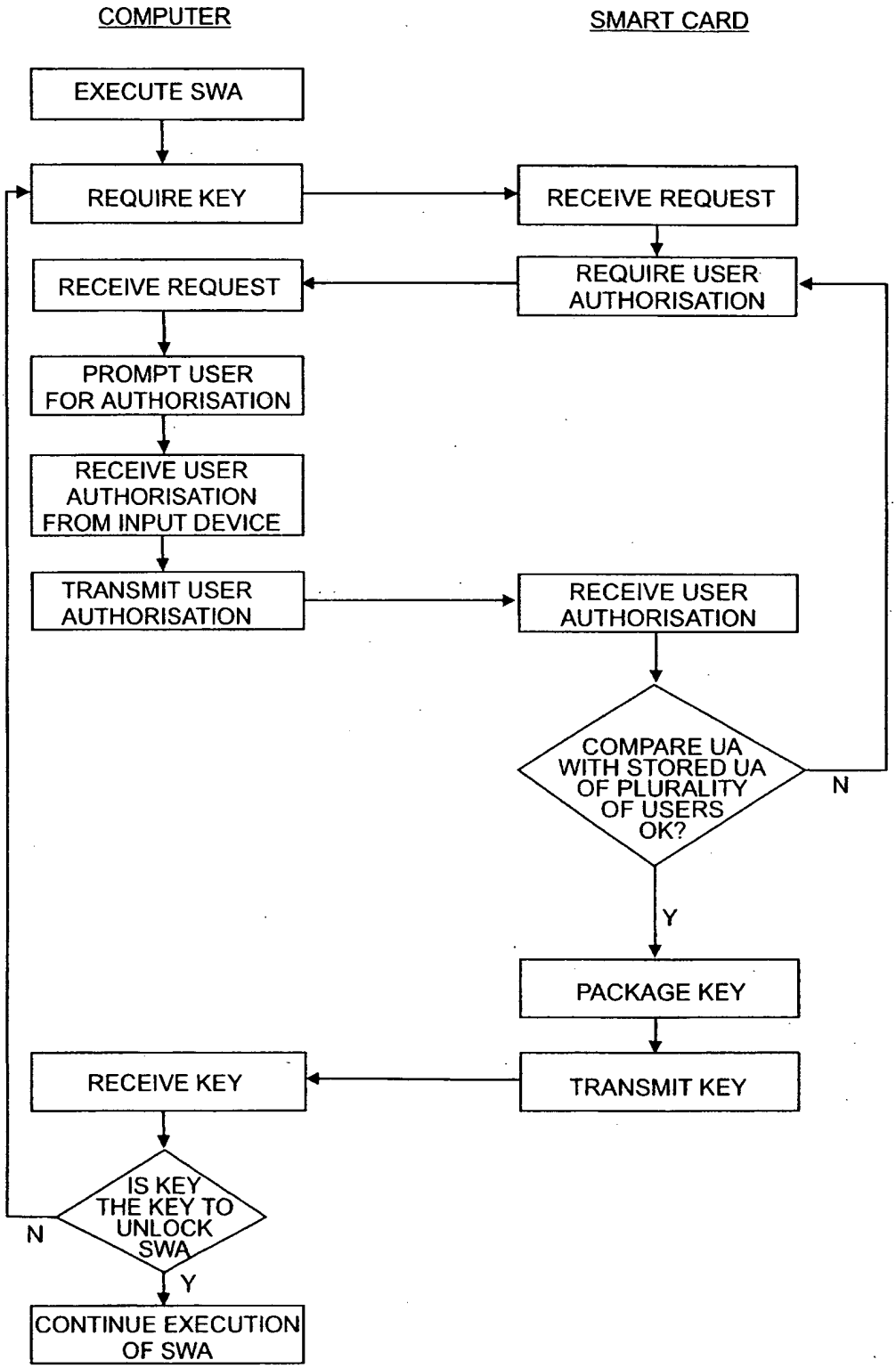

COMPUTER
SMART CARD
EXECUTE SWA
REQUIRE KEY
RECEIVE REQUEST
REQUIRE USER AUTHORISATION
RECEIVE REQUEST
PROMPT USER FOR AUTHORISATION
RECEIVE USER AUTHORISATION FROM INPUT DEVICE
TRANSMIT USER AUTHORISATION
RECEIVE USER AUTHORISATION
COMPARE UA WITH STORED UA OF PLURALITY OF USERS OK?
N
Y
PACKAGE KEY
TRANSMIT KEY
RECEIVE KEY
IS KEY THE KEY TO UNLOCK SWA
N
Y
CONTINUE EXECUTION OF SWA

Figure 6

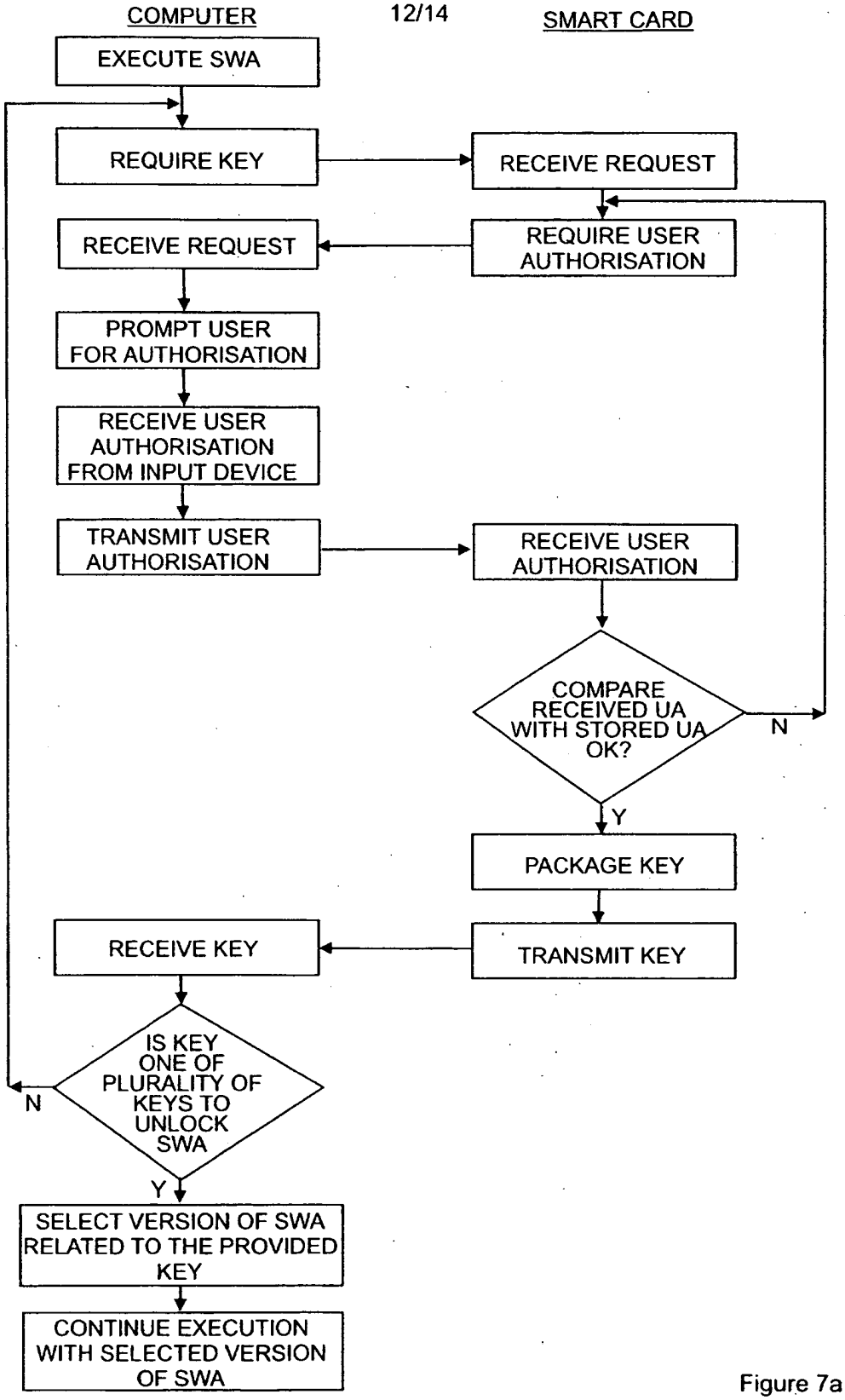

COMPUTER
SMART CARD
EXECUTE SWA
REQUIRE KEY
RECEIVE REQUEST
REQUIRE USER AUTHORISATION
RECEIVE REQUEST
PROMPT USER FOR AUTHORISATION
RECEIVE USER AUTHORISATION FROM INPUT DEVICE
TRANSMIT USER AUTHORISATION
RECEIVE USER AUTHORISATION
COMPARE RECEIVED UA WITH STORED UA OK?
N
Y
PACKAGE KEY
TRANSMIT KEY
RECEIVE KEY
IS KEY ONE OF PLURALITY OF KEYS TO UNLOCK SWA
N
Y
SELECT VERSION OF SWA RELATED TO THE PROVIDED KEY
CONTINUE EXECUTION WITH SELECTED VERSION OF SWA

Figure 7a

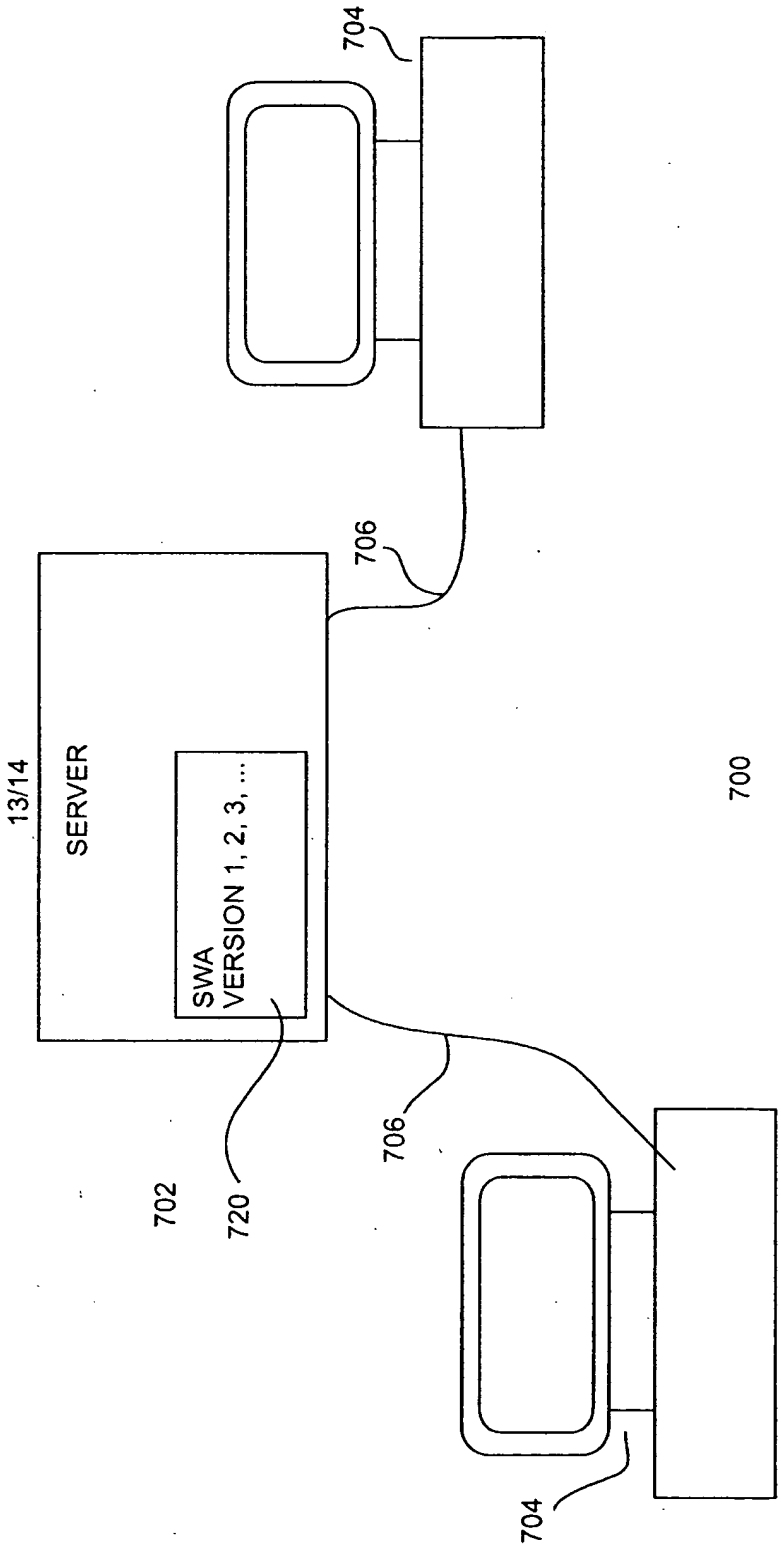


Figure 7b

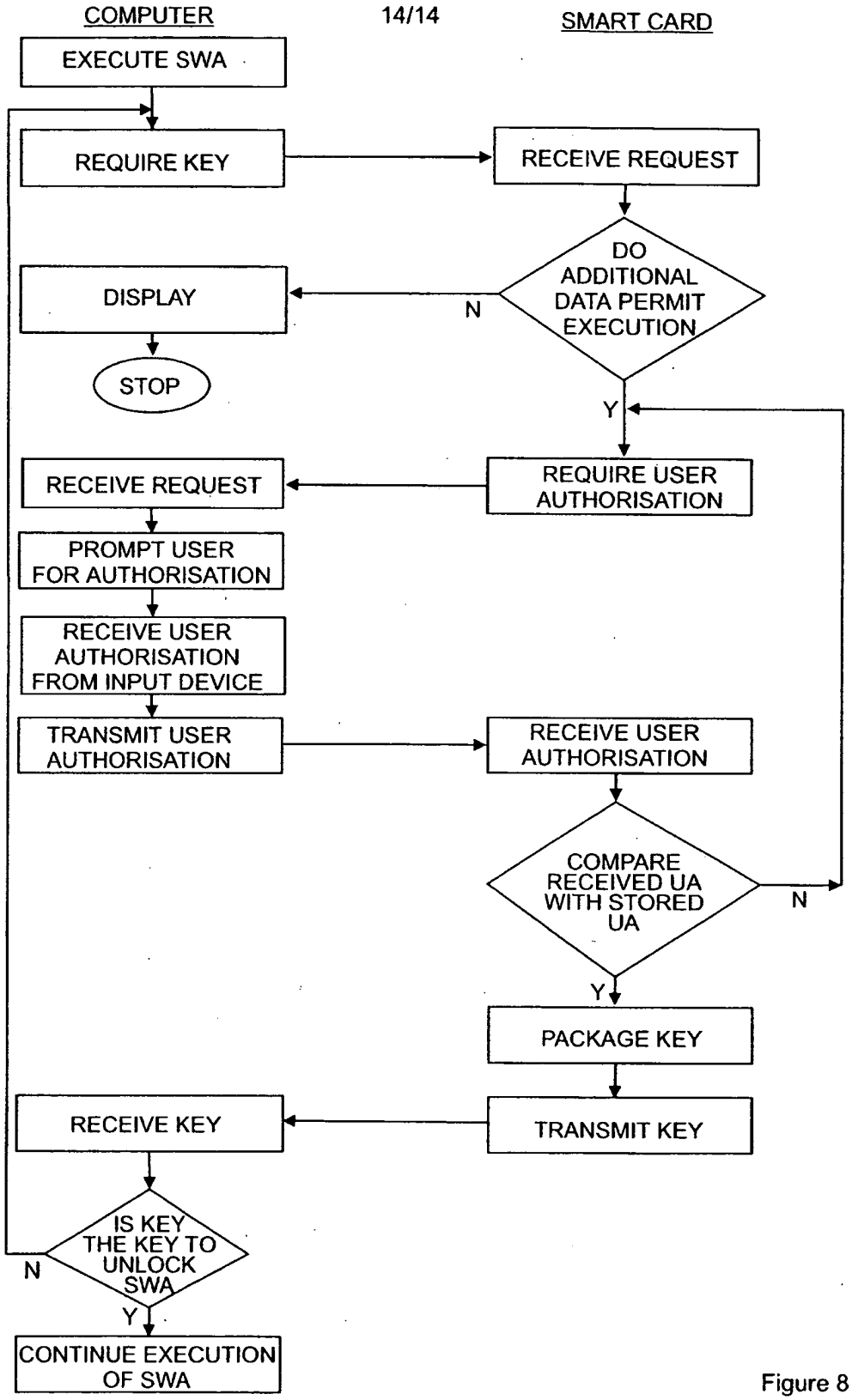

COMPUTER
SMART CARD
EXECUTE SWA
REQUIRE KEY
RECEIVE REQUEST
DO ADDITIONAL DATA PERMIT EXECUTION
N
DISPLAY
STOP
Y
REQUIRE USER AUTHORISATION
RECEIVE REQUEST
PROMPT USER FOR AUTHORISATION
RECEIVE USER AUTHORISATION FROM INPUT DEVICE
TRANSMIT USER AUTHORISATION
RECEIVE USER AUTHORISATION
COMPARE RECEIVED UA WITH STORED UA
N
Y
PACKAGE KEY
TRANSMIT KEY
RECEIVE KEY
IS KEY THE KEY TO UNLOCK SWA
N
Y
CONTINUE EXECUTION OF SWA

Figure 8